

FIG. 1

Send, from TIU_0 to GC_0 and GC_T , a setup message for the call ~210

Establish a gate at NED_T upon receiving the setup message from GC_T ~220

Establish a gate at NED_0 upon receiving the setup message from GC_0 ~230

Send a reserve message from TIU_0 to NED_0 ~240

Send a reserve message from TIU_T to NED_T ~250

Exchange end-to-end message between TIU_0 and TIU_T ~260

Upon connecting the calling party and the called party, send a commit message from TIU_0 to NED_0 and from TIU_T to NED_T ~270

Upon receiving the commit message at NED_0 , open the gate at NED_0 ~280

Upon receiving the commit message at NED_T , open the gate at NED_T ~290

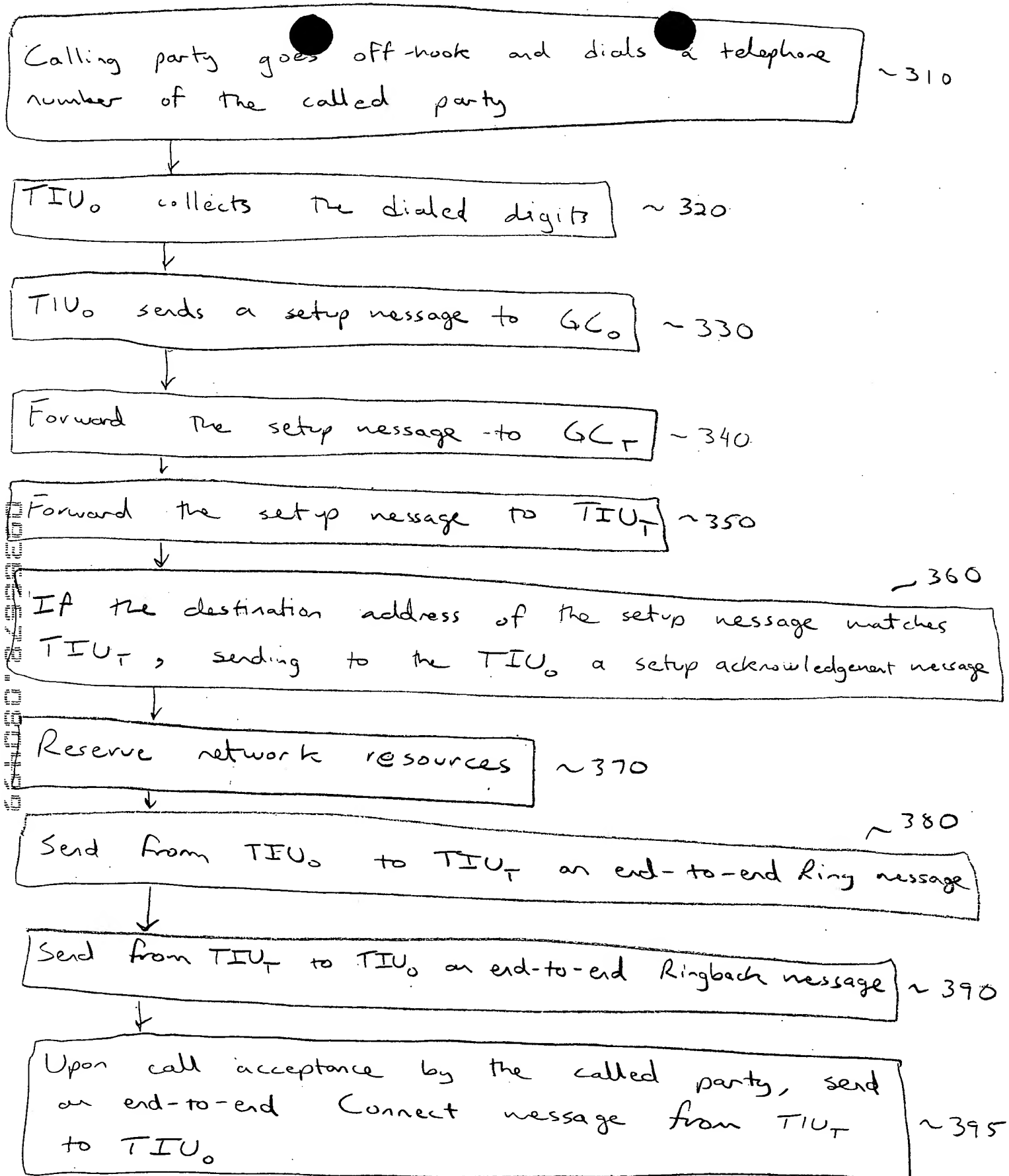


FIG. 3

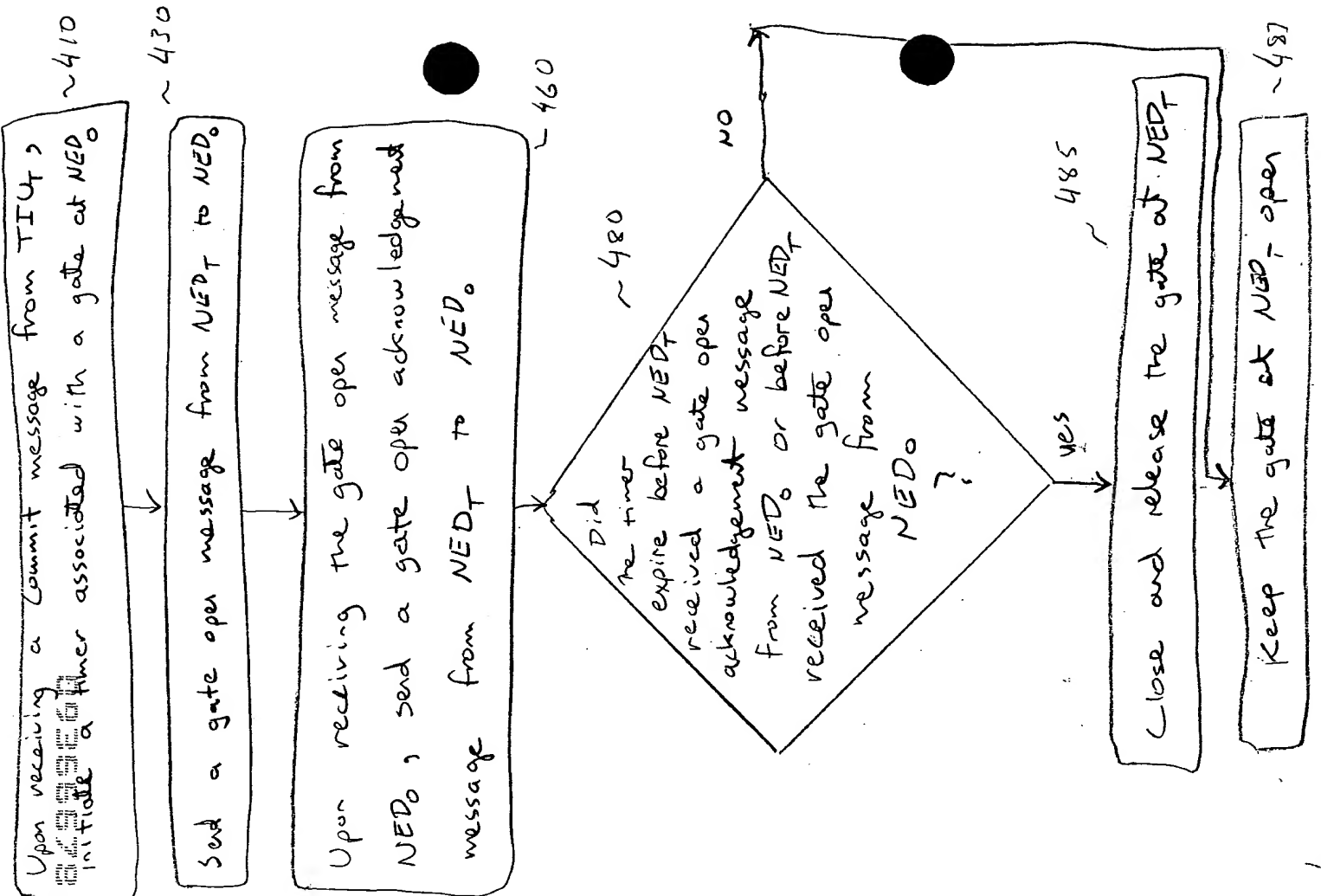
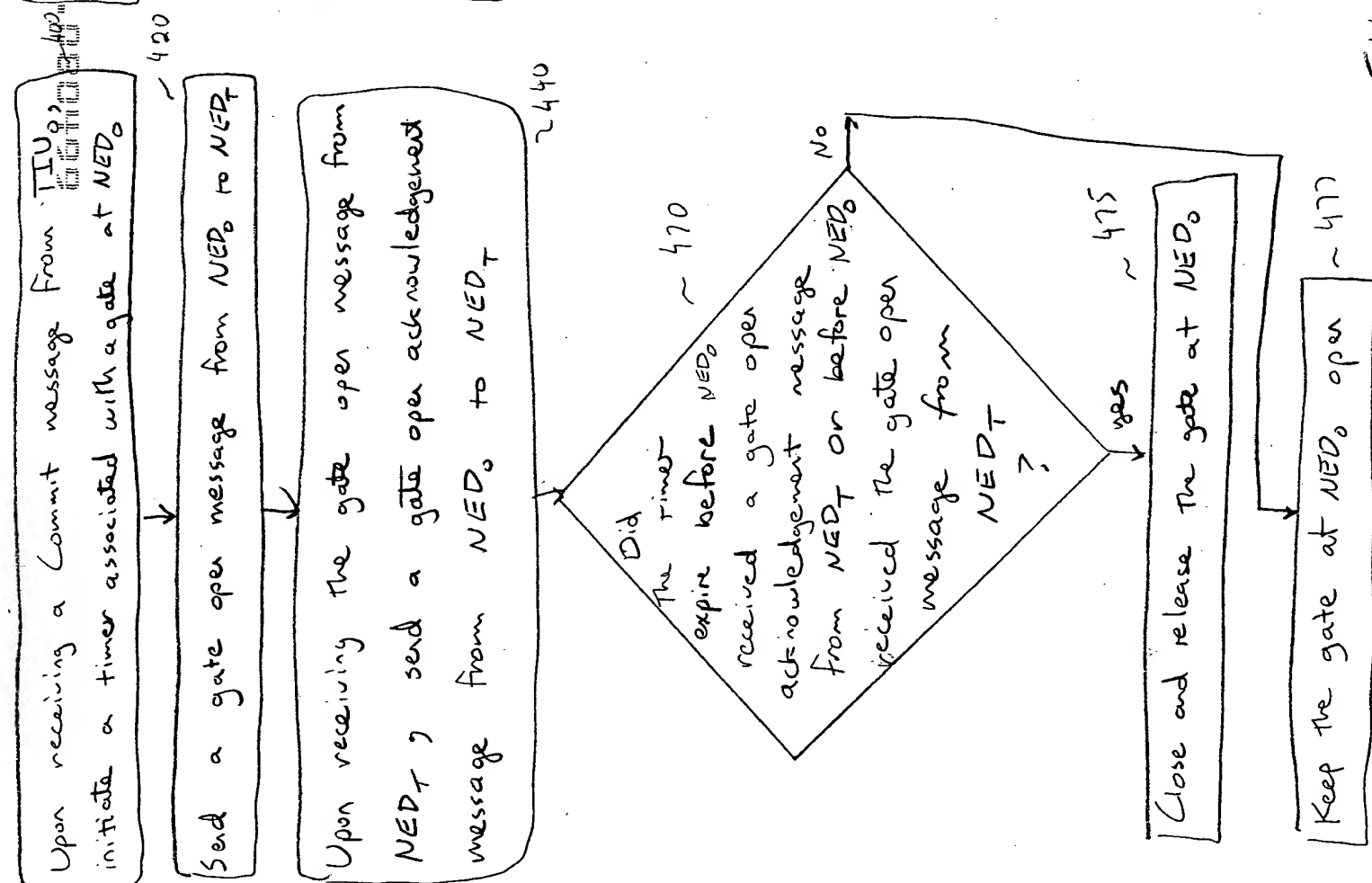


FIG. 4

Packets are sent from TIU₀ to NED₀ ~ 500



Translate the local source address and local destination address to a global source address and a global destination address ~510



Forward the translated packets from NED_0 to NED_T



Translate the global source address and the global destination address to a second local source address and a second local address ~530



Send the translated packets from NED_T to TIU_T ~ 590

Flu. 5

```

sequenceDiagram
    participant BTI_O as BTI_O
    participant ER_O as ER_O
    participant GC_O as GC_O
    participant GC_T as GC_T
    participant ER_T as ER_T
    participant BTI_T as BTI_T

    Note over BTI_O: Provides dialtone.  
Collects complete E.164_T.
    BTI_O->>GC_O: SETUP
    Note over GC_O: Authenticates BTI_O via AI. Matches  
E.164_T to GC_T. Determines there is  
sufficient network capacity for call.
    GC_O->>ER_O: GATEALLOC
    Note over ER_O: Allocates the  
gate for this call.
    ER_O->>GC_O: GATEALLOCACK
    GC_O->>GC_T: GCSETUP
    Note over GC_T: Translates E.164_T to BTI_T and  
matches with ER_T.
    GC_T->>ER_T: GATESETUP
    Note over ER_T: "Establishes the gate" for this call.
    ER_T->>GC_T: GATESETUPACK
    GC_T->>BTI_T: SETUP
    BTI_T->>GC_T: SETUPACK
    GC_T->>GC_O: GCSETUPACK
    GC_O->>ER_O: GATESETUP
    Note over ER_O: "Establishes the gate" for this call.
    ER_O->>GC_O: GATESETUPACK
    GC_O->>BTI_O: SETUPACK
    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Additional Capability Negotiation (if necessary)
    BTI_O->>ER_O: RESERVE
    Note over ER_O, GC_O, GC_T, ER_T: Intermediate Routers  
Reserves Backbone Channel
    ER_T->>BTI_T: RESERVE
    ER_O->>BTI_O: RESERVEVEACK
    BTI_T->>ER_T: RESERVEVEACK
    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: RING
    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: RINGBACK
    Note over BTI_O: Starts ringback
    Note over ER_T, BTI_T: Rings phone
    Note over ER_T, BTI_T: Detects off-hook
    BTI_O->>ER_O: COMMIT
    ER_T->>BTI_T: COMMIT
    ER_O->>GC_O: GATEOPEN
    Note over ER_O: Allocates access channel.  
Generates call record.
    GC_O->>ER_T: GATEOPEN
    Note over GC_O: Allocates access channel.  
Generates call record.
    ER_T->>GC_O: GATEOPENACK
    GC_O->>ER_O: GATEOPENACK
    ER_O->>BTI_O: COMMITACK
    BTI_T->>ER_T: COMMITACK
    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Call In Progress
  
```

The diagram illustrates the ISDN call setup process between the originating and terminating entities (BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T) and the intermediate routers. The process begins with BTI_O providing a dialtone and collecting the complete E.164_T number. It then sends a SETUP message to GC_O. GC_O authenticates BTI_O via AI, matches E.164_T to GC_T, and determines if there is sufficient network capacity for the call. GC_O then sends a GATEALLOC message to ER_O, which allocates the gate for this call and responds with GATEALLOCACK. GC_O sends GCSETUP to GC_T, which translates E.164_T to BTI_T and matches it with ER_T. GC_T sends GATESETUP to ER_T, which establishes the gate for this call and responds with GATESETUPACK. GC_T then sends SETUP to BTI_T, which responds with SETUPACK. GC_T sends GCSETUPACK to GC_O, which sends GATESETUP to ER_O, which establishes the gate for this call and responds with GATESETUPACK. GC_O then sends SETUPACK to BTI_O. An additional capability negotiation (if necessary) follows. The process then moves to the reservation phase, where BTI_O sends RESERVE to ER_O, and ER_T sends RESERVE to BTI_T. Intermediate routers reserve the backbone channel. ER_O sends RESERVEVEACK to BTI_O, and BTI_T sends RESERVEVEACK to ER_T. The process then moves to the ringing phase, where BTI_O starts ringback and ER_T rings the phone. BTI_O sends COMMIT to ER_O, and ER_T sends COMMIT to BTI_T. ER_O sends GATEOPEN to GC_O, which allocates an access channel and generates a call record. GC_O sends GATEOPEN to ER_T, which also allocates an access channel and generates a call record. ER_T sends GATEOPENACK to GC_O, which sends GATEOPENACK to ER_O. ER_O sends COMMITACK to BTI_O, and BTI_T sends COMMITACK to ER_T. The process ends with the call in progress.

Figure 6

```

sequenceDiagram
    participant BTI_O
    participant ER_O
    participant BR1 as Backbone Router
    participant BR2 as Backbone Router
    participant ER_T
    participant BTI_T

    BTI_O->>ER_O: RESERVE
    Note over ER_O: Checks that GID_O is set up. Looks up GA_O and GA_T. Checks that upstream capacity is available in access and reserves. Reserves capacity in backbone network.
    ER_O->>BR1: BACKBONERESERVE
    Note over BR1: Checks that forward capacity is available and reserves. Forwards reservation message.
    BR1->>BR2: BACKBONERESERVE
    Note over BR2: Checks that forward capacity is available and reserves. Forwards reservation message.
    BR2->>ER_T: BACKBONERESERVE
    Note over ER_T: Matches GA_O, PN_O, GA_T, PN_T with gate GID_T.
    ER_T->>BR2: RESERVEACK
    BR2->>BR1: BACKBONERESERVEACK
    BR1->>ER_O: BACKBONERESERVEACK
    ER_O->>BTI_O: RESERVE-ACK (TID)
  
```

The diagram illustrates the reservation protocol between the following entities: **BTI_O**, **ER_O**, **Backbone Router** (two instances), **ER_T**, and **BTI_T**.

- BTI_O** sends a **RESERVE** message to **ER_O**.
- ER_O** performs a check: "Checks that GID_O is set up. Looks up GA_O and GA_T. Checks that upstream capacity is available in access and reserves. Reserves capacity in backbone network."
- ER_O** sends a **BACKBONERESERVE** message to the first **Backbone Router**.
- The first **Backbone Router** performs a check: "Checks that forward capacity is available and reserves. Forwards reservation message."
- The first **Backbone Router** sends a **BACKBONERESERVE** message to the second **Backbone Router**.
- The second **Backbone Router** performs a check: "Checks that forward capacity is available and reserves. Forwards reservation message."
- The second **Backbone Router** sends a **BACKBONERESERVE** message to **ER_T**.
- ER_T** performs a check: "Matches GA_O, PN_O, GA_T, PN_T with gate GID_T."
- ER_T** sends a **RESERVEACK** message to the second **Backbone Router**.
- The second **Backbone Router** sends a **BACKBONERESERVEACK** message to the first **Backbone Router**.
- The first **Backbone Router** sends a **BACKBONERESERVEACK** message to **ER_O**.
- ER_O** sends a **RESERVE-ACK (TID)** message to **BTI_O**.

Figure 7

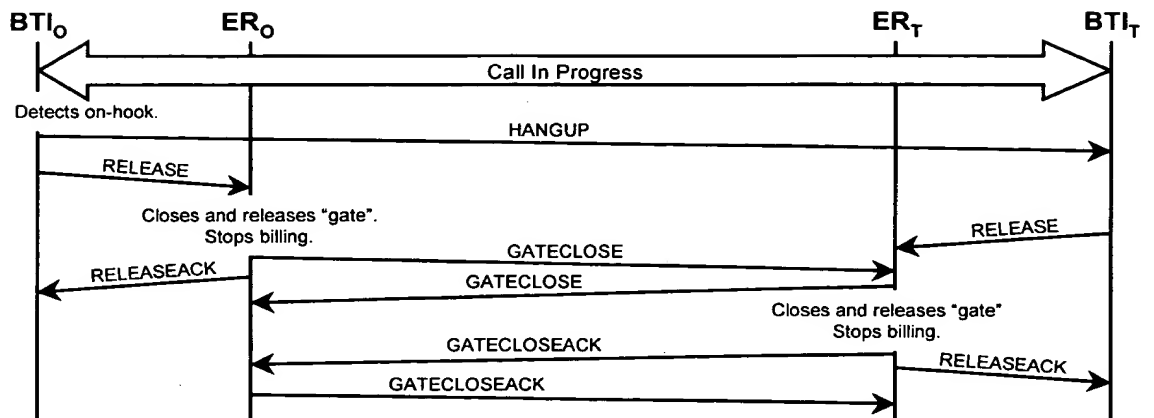


Figure 8

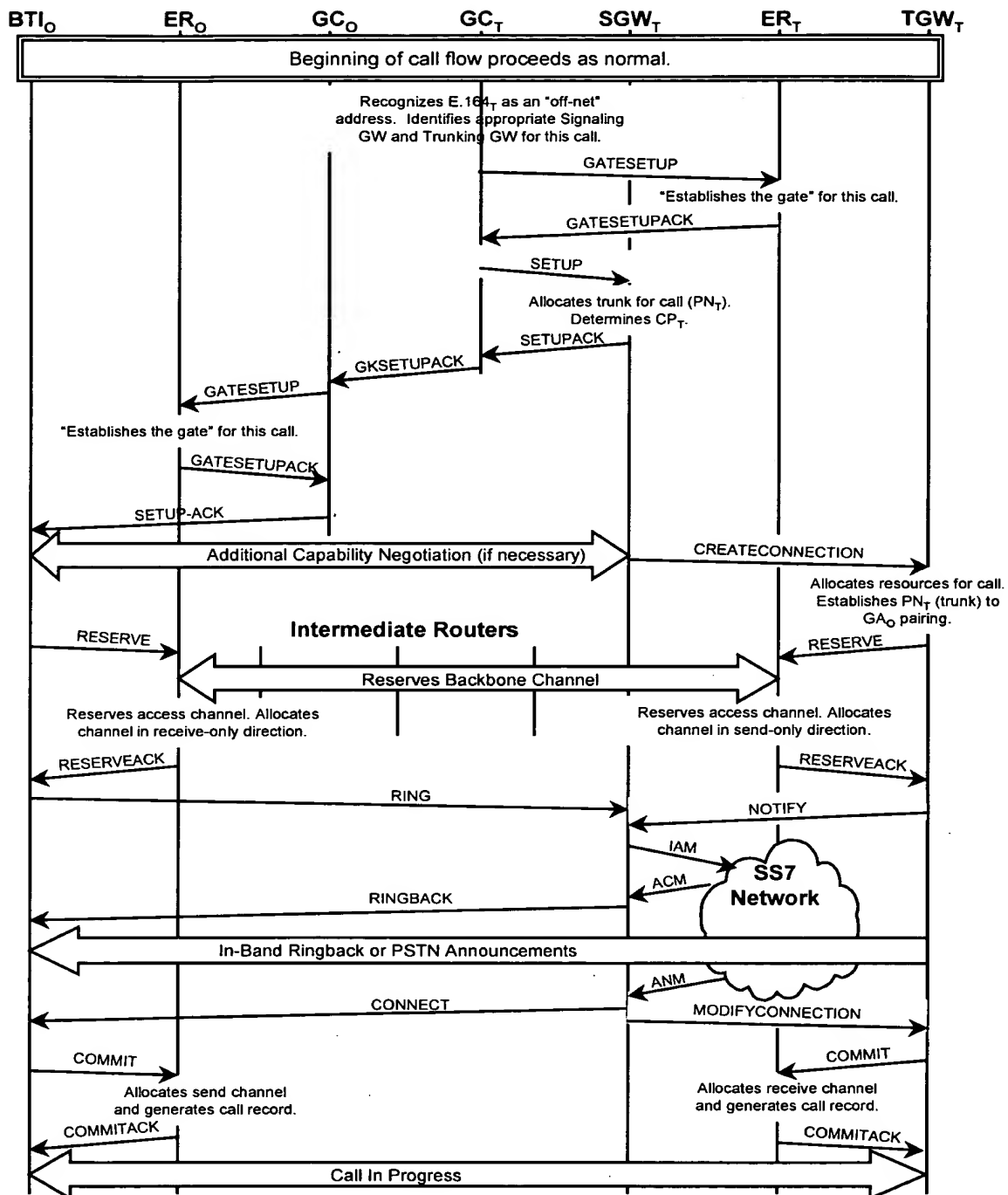


Figure 9

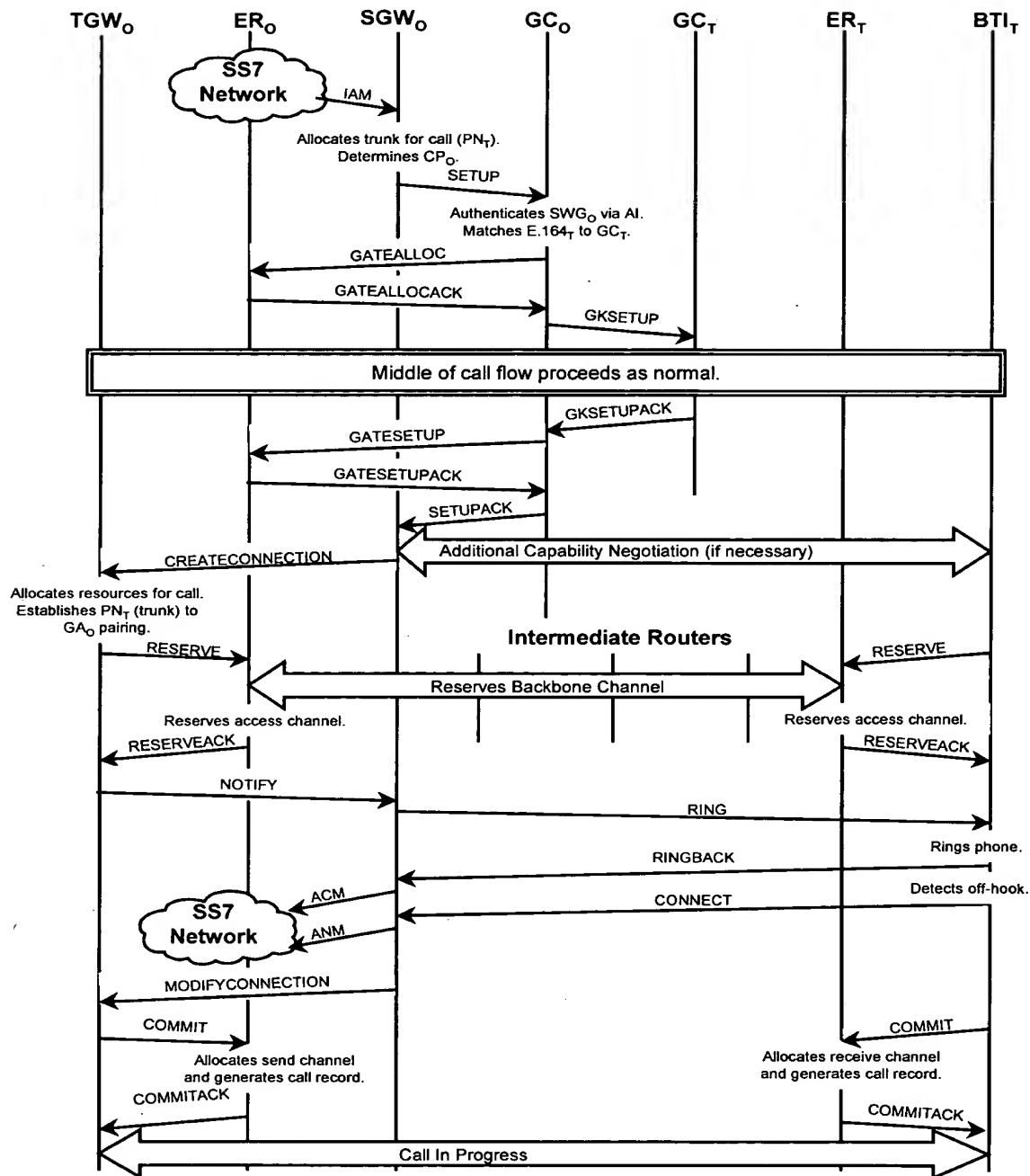


Figure 10

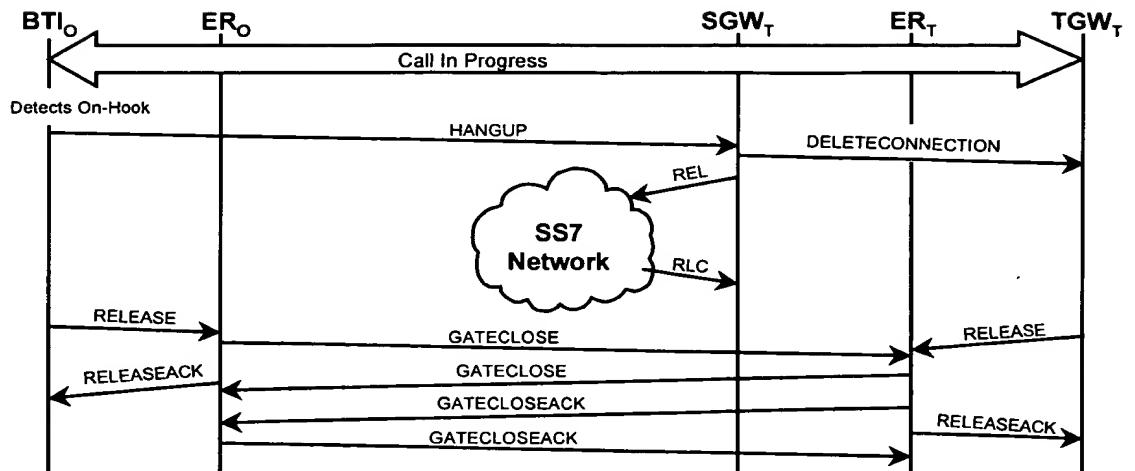


Figure 11

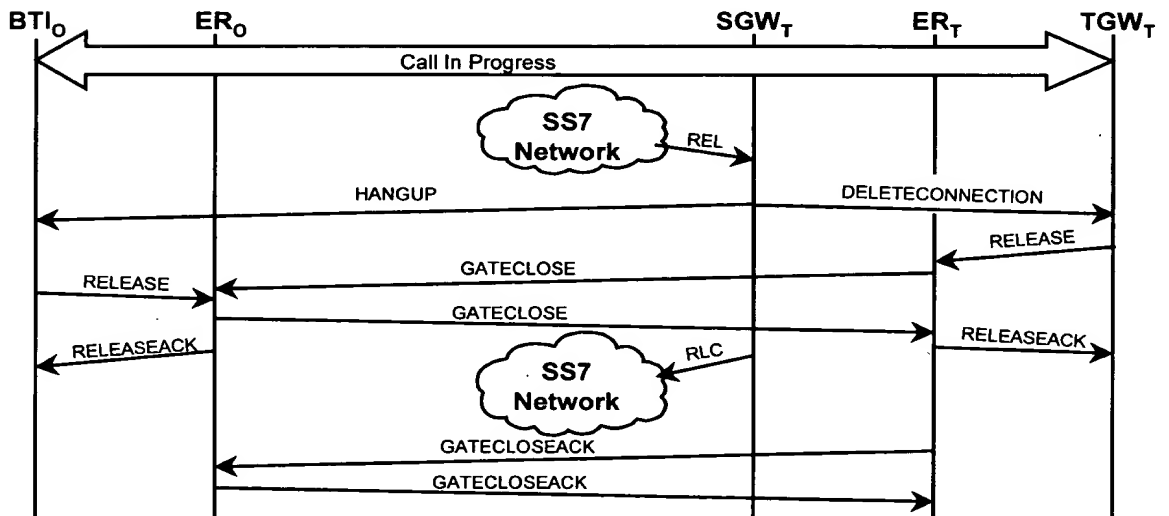


Figure 12

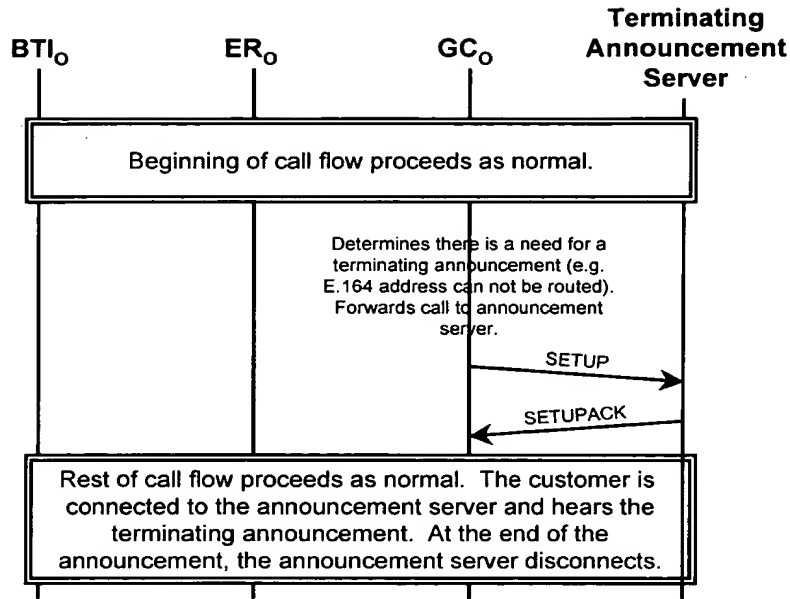


Figure 13

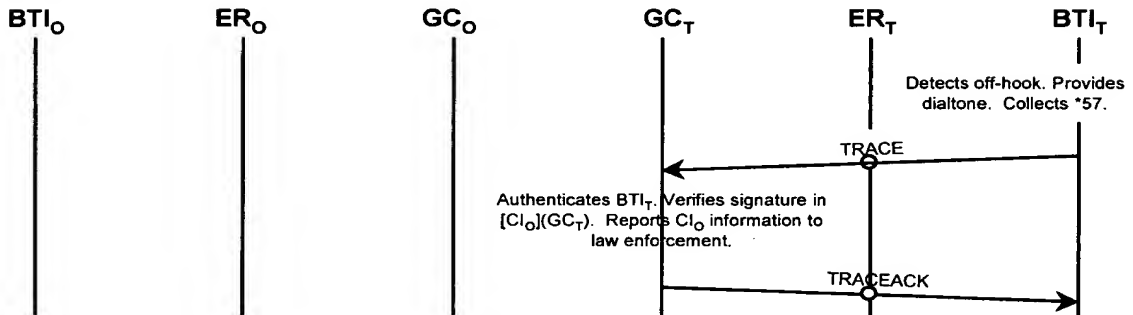


Figure 14

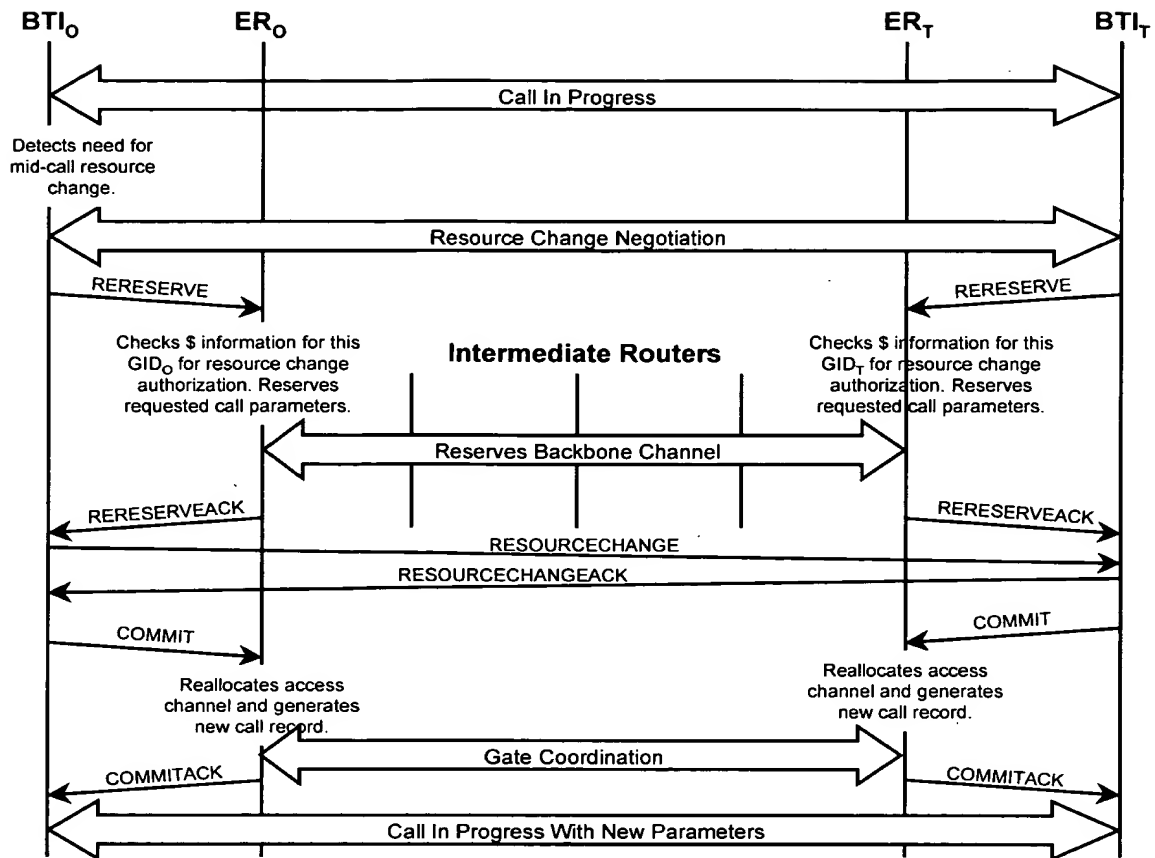


Figure 15

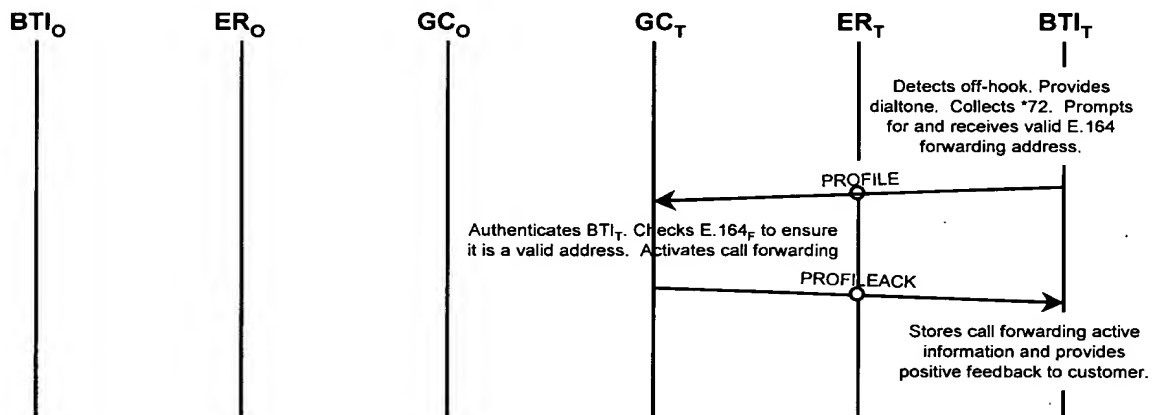


Figure 16

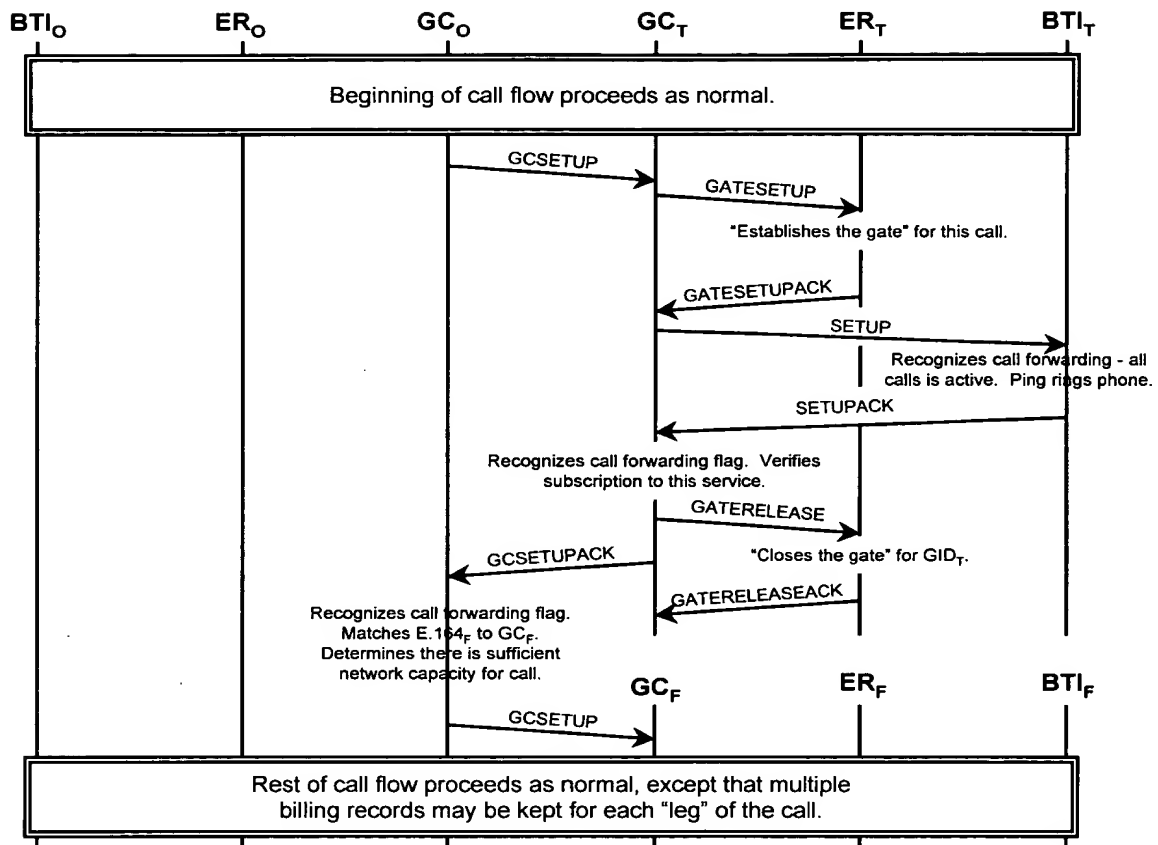


Figure 17

Copyright © 2000 by Cisco Systems, Inc.

```

sequenceDiagram
    participant BTI_O
    participant ER_O
    participant GC_O
    participant GC_T
    participant ER_T
    participant BTI_T
    participant GC_F
    participant ER_F
    participant BTI_F

    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Beginning of call flow proceeds as normal.

    GC_O->>GC_T: GCSETUP
    GC_T->>ER_T: GATESETUP
    Note over ER_T: "Establishes the gate" for this call.
    ER_T->>GC_T: GATESETUPACK
    ER_T->>BTI_T: SETUP
    Note over BTI_T: X BTI_T out of service.
    GC_T->>ER_T: GATERELEASE
    Note over ER_T: "Closes the gate" for GID_T.
    ER_T->>GC_T: GATERELEASEACK
    GC_O->>GC_T: GCSETUPACK
    GC_O->>GC_F: GCSETUP
    Note over GC_O: Recognizes call forwarding flag.  
Matches E.164_F to GC_F.  
Determines there is sufficient  
network capacity for call.

    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Rest of call flow proceeds as normal, except that multiple  
billing records may be kept for each "leg" of the call.
  
```

Figure 18

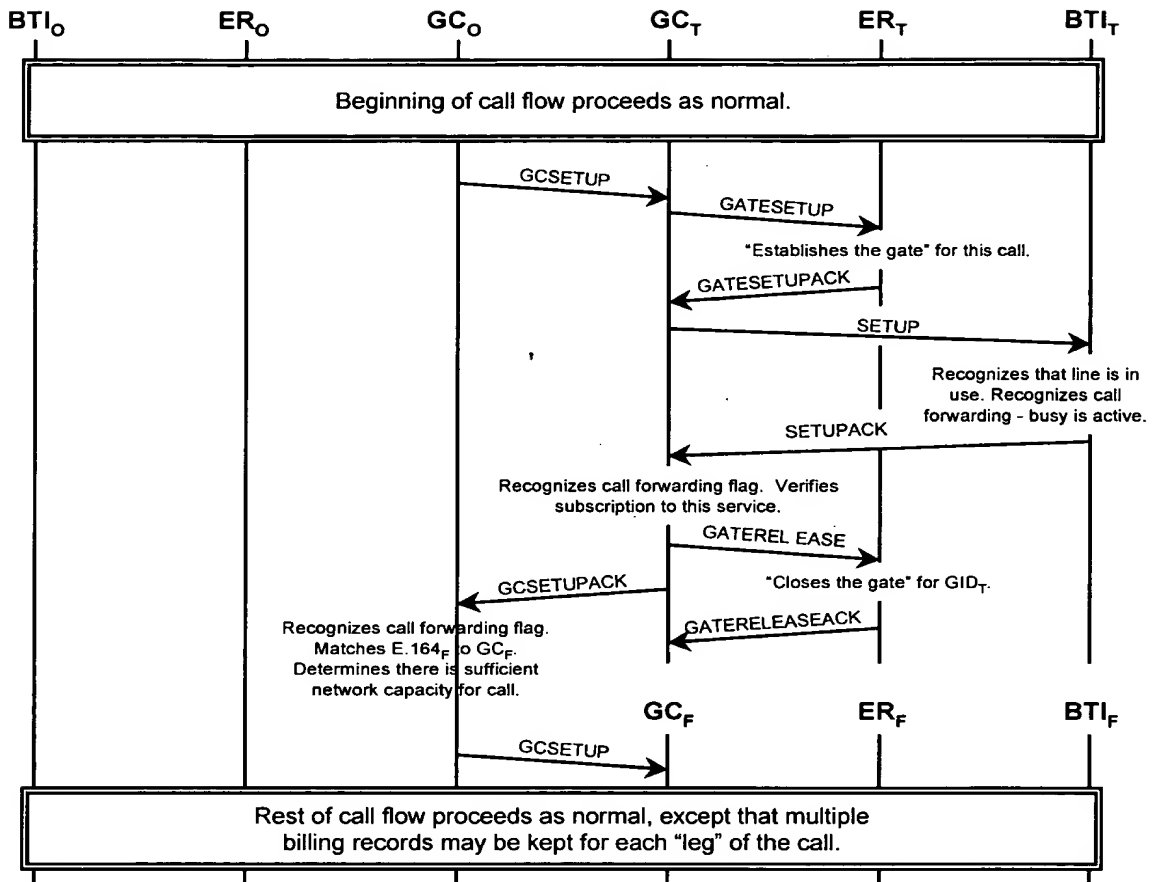


Figure 19

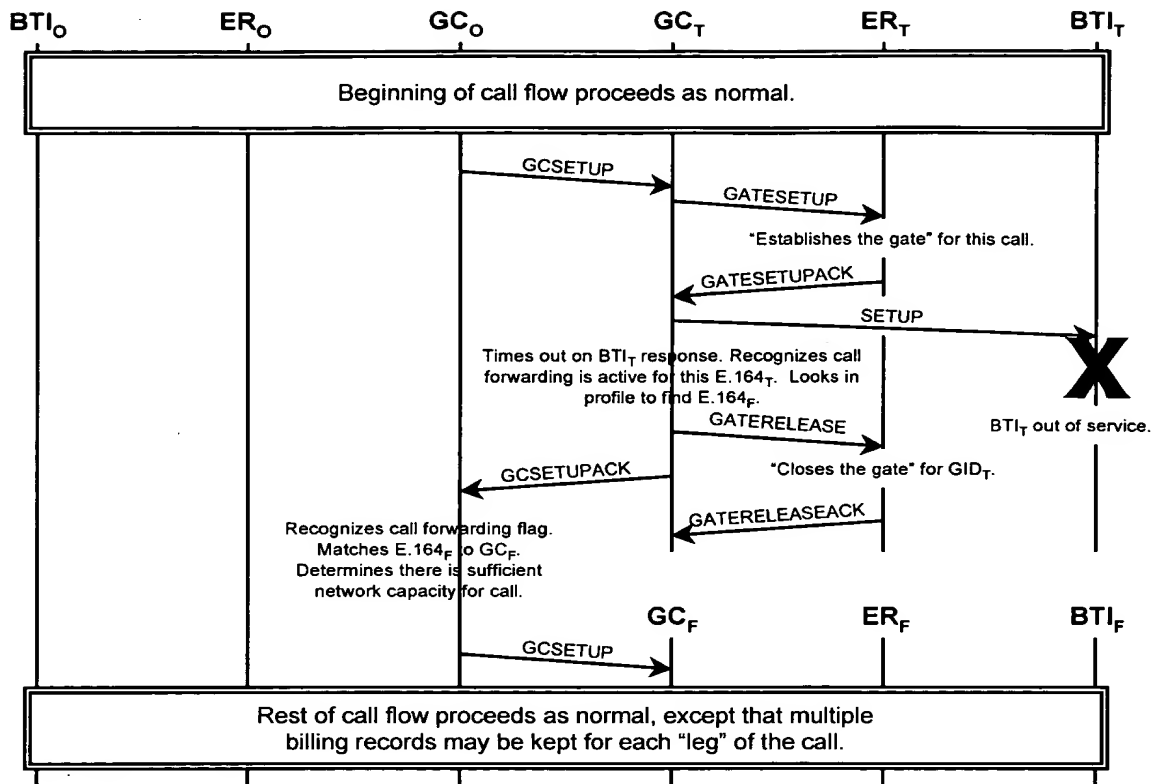


Figure 20

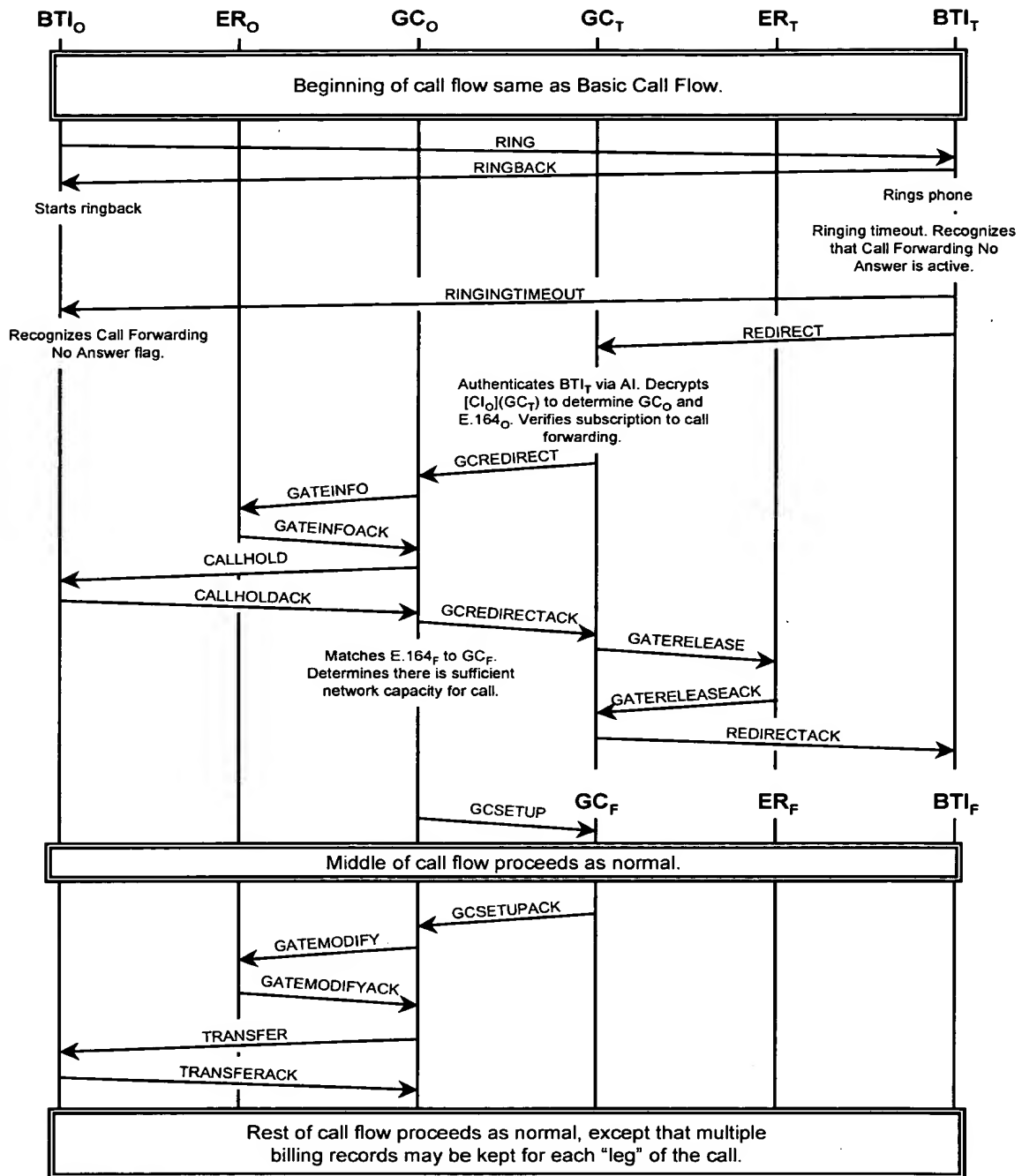


Figure 21

```

sequenceDiagram
    participant BTI_O
    participant ER_O
    participant GC_O
    participant GC_T
    participant ER_T
    participant BTI_T
    participant GC_F
    participant ER_F
    participant BTI_F

    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Beginning of call flow proceeds as normal.
    GC_O->>GC_T: GCSETUP
    GC_T->>ER_T: GATESETUP
    ER_T-->>GC_T: GATESETUPACK
    ER_T->>BTI_T: SETUP
    Note over BTI_T: BTI_T out of service.
    Note over GC_T: Times out on BTI_T response. Recognizes  
call forwarding is active for this E.164_T.  
Looks in profile to find E.164_F.
    GC_T-->>GC_O: GCSETUPACK
    Note over GC_O: Recognizes call forwarding flag.  
Matches E.164_F to GC_F.  
Determines there is sufficient  
network capacity for call.
    GC_O->>GC_F: GCSETUP
    GC_T->>ER_T: GATERELEASE
    ER_T-->>GC_T: GATERELEASEACK
    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T, GC_F, ER_F, BTI_F: Rest of call flow proceeds as normal, except that multiple  
billing records may be kept for each "leg" of the call.
  
```

Figure 22

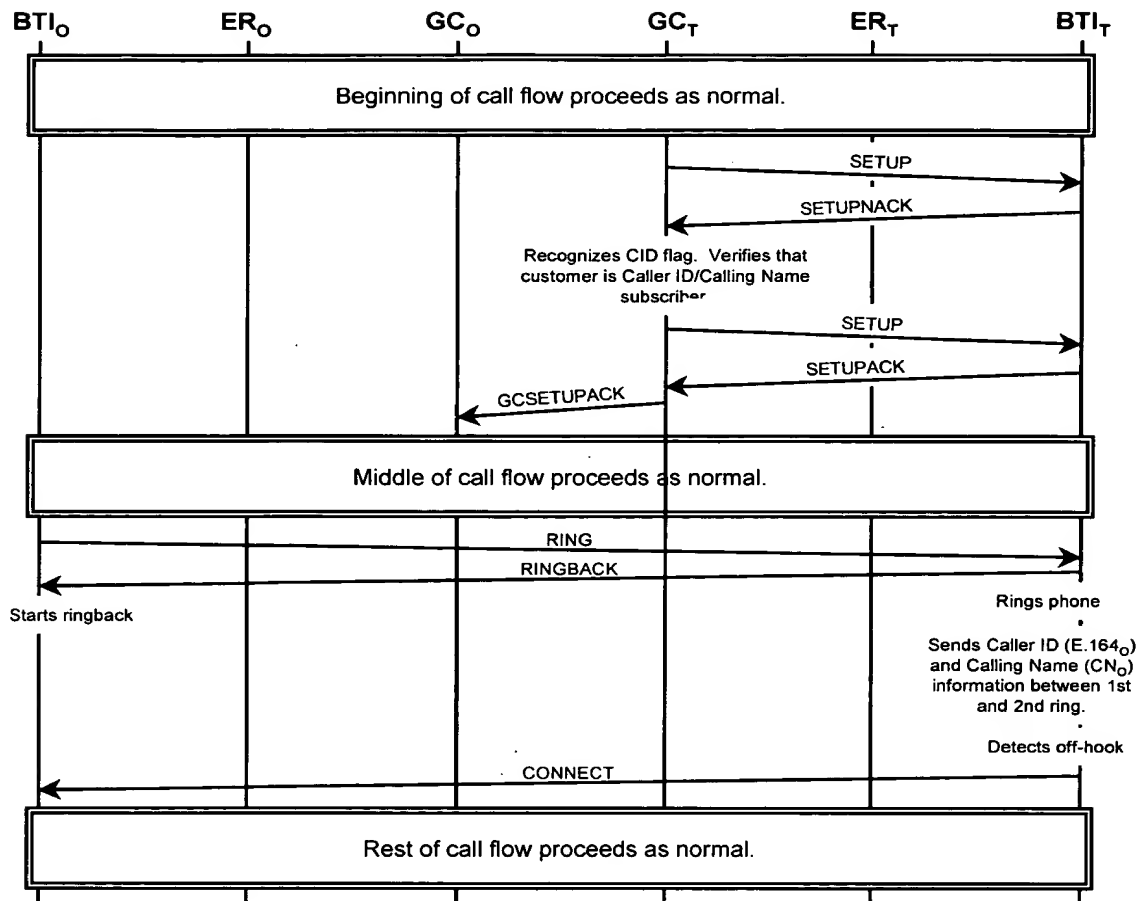


Figure 23

64000 64000 64000

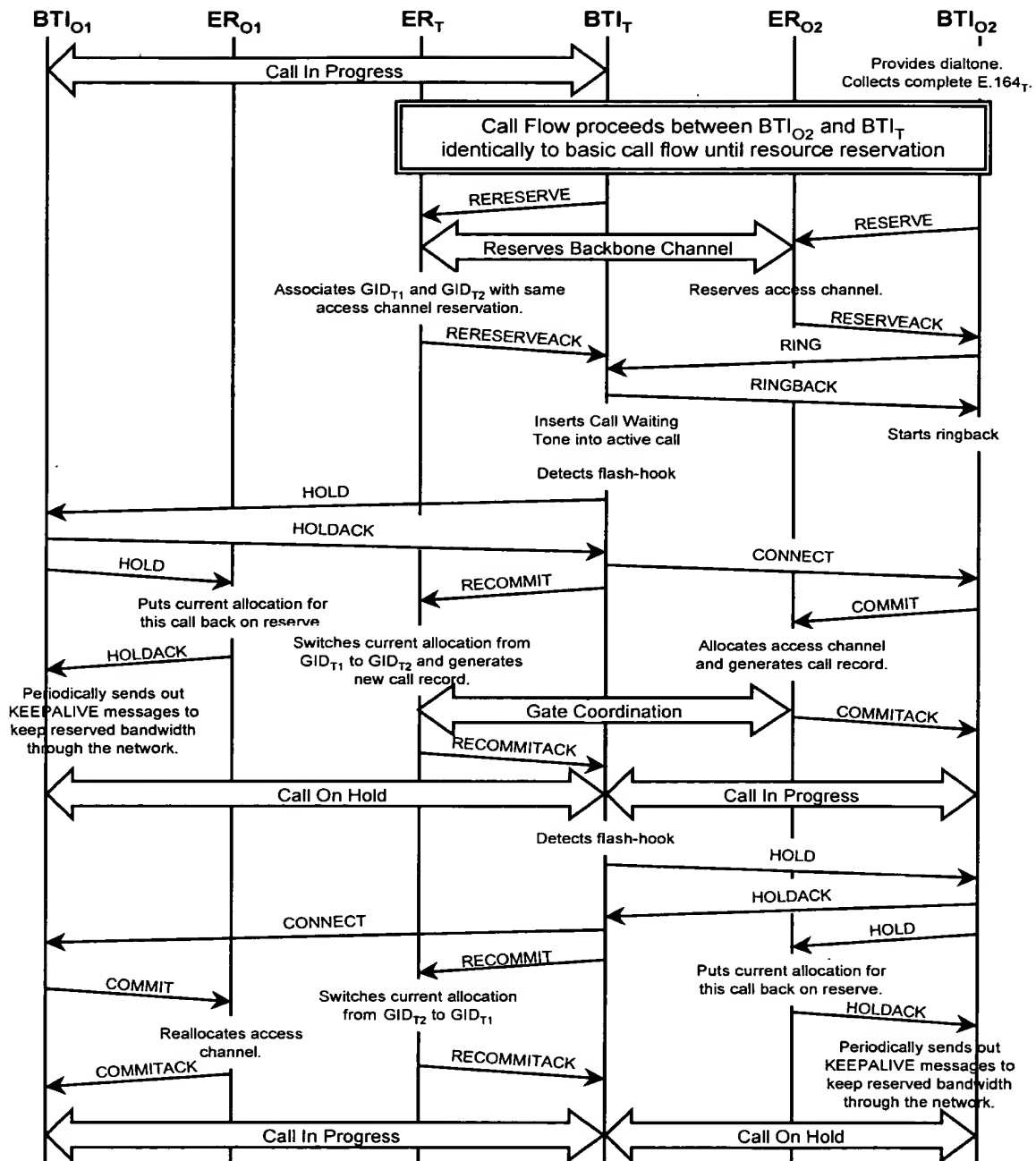


Figure 24

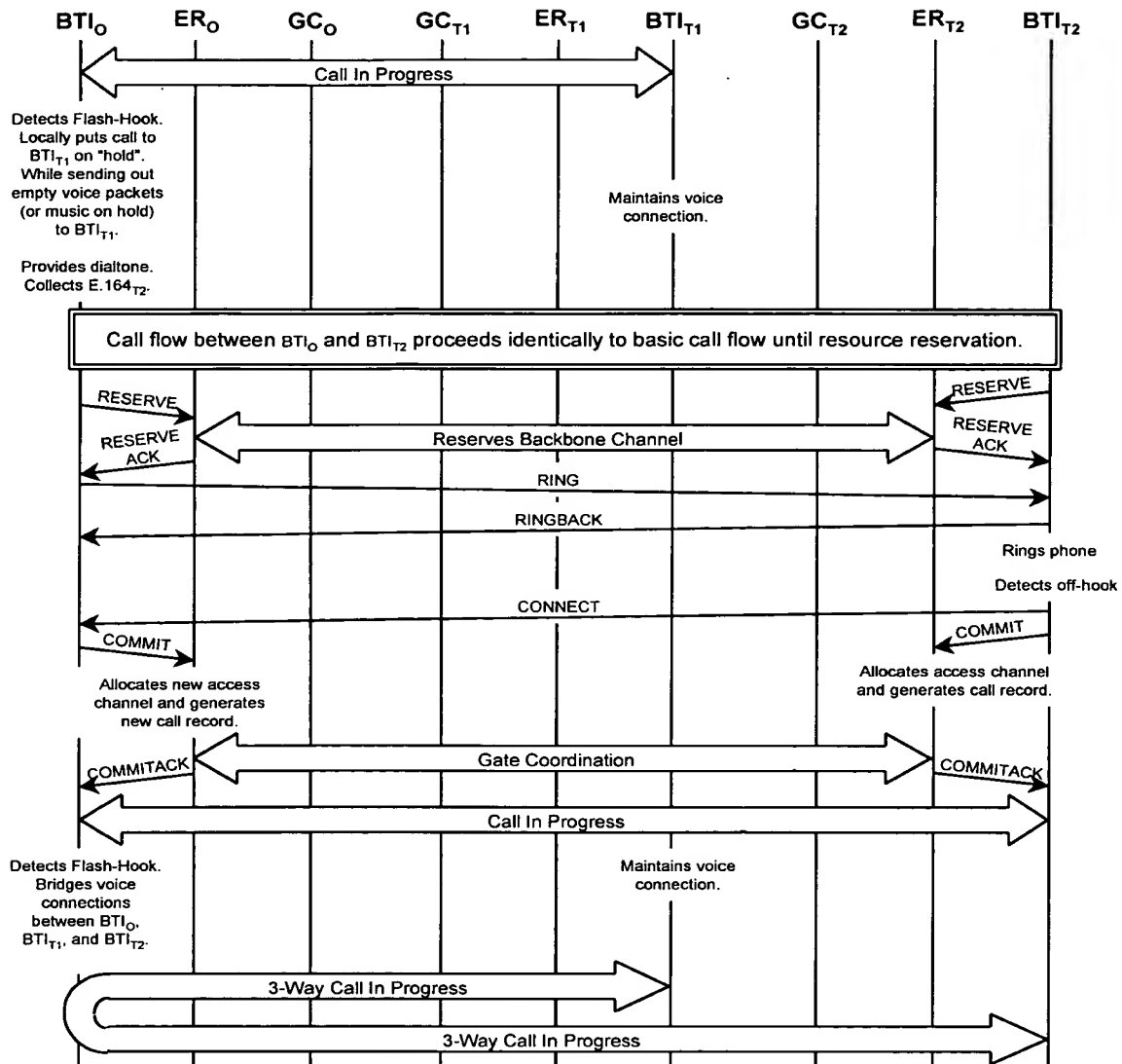


Figure 25

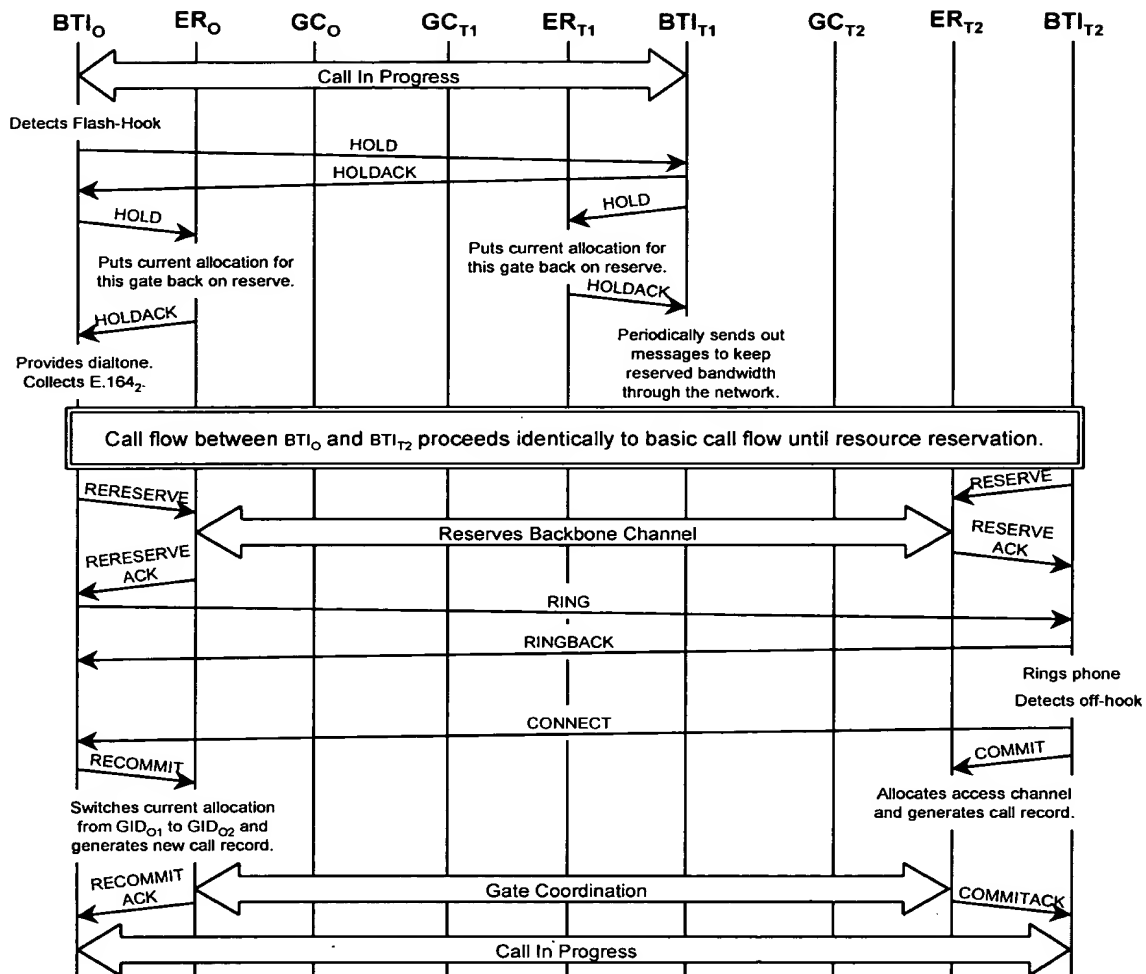


Figure 26

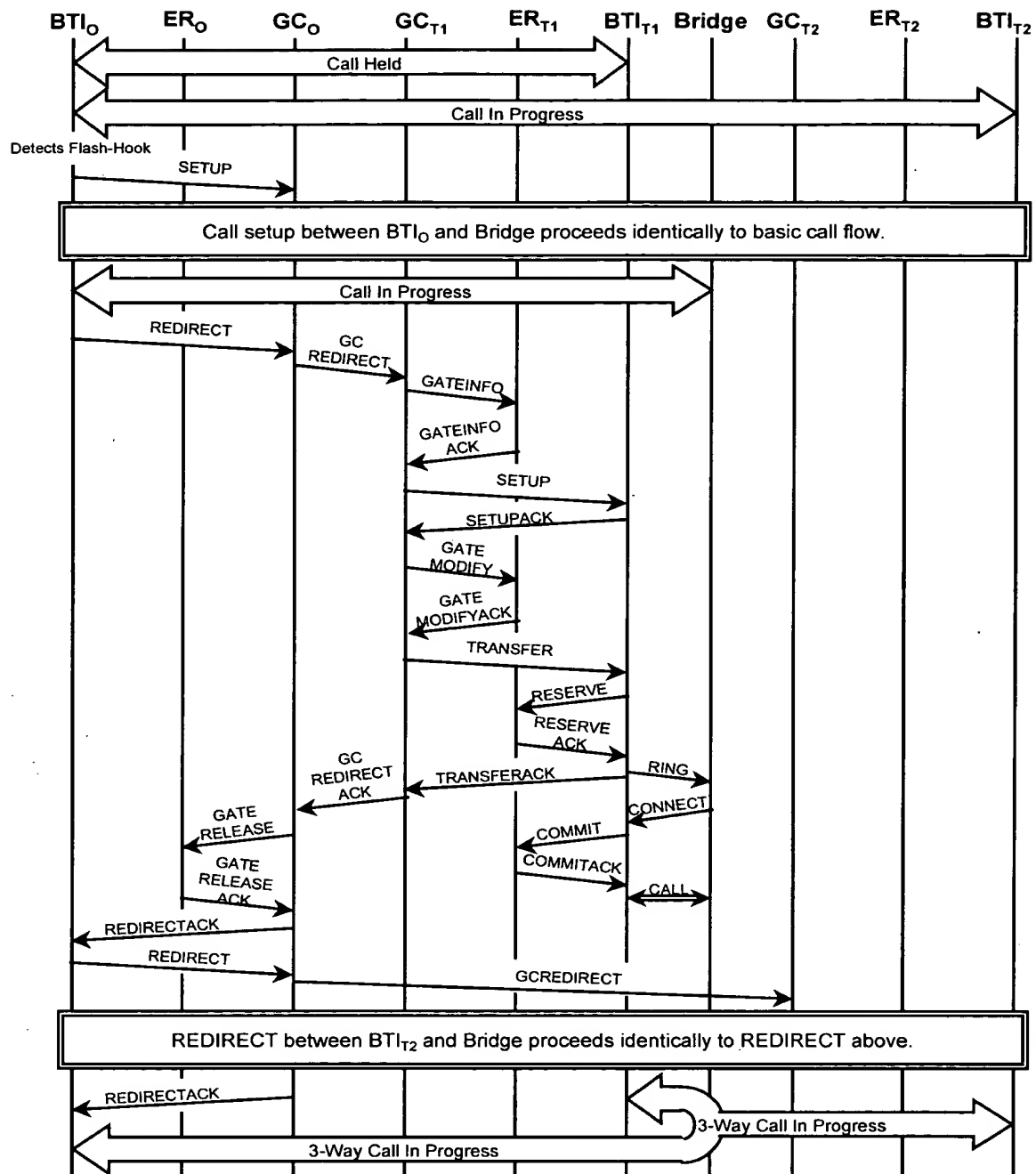


Figure 27

661100 000000

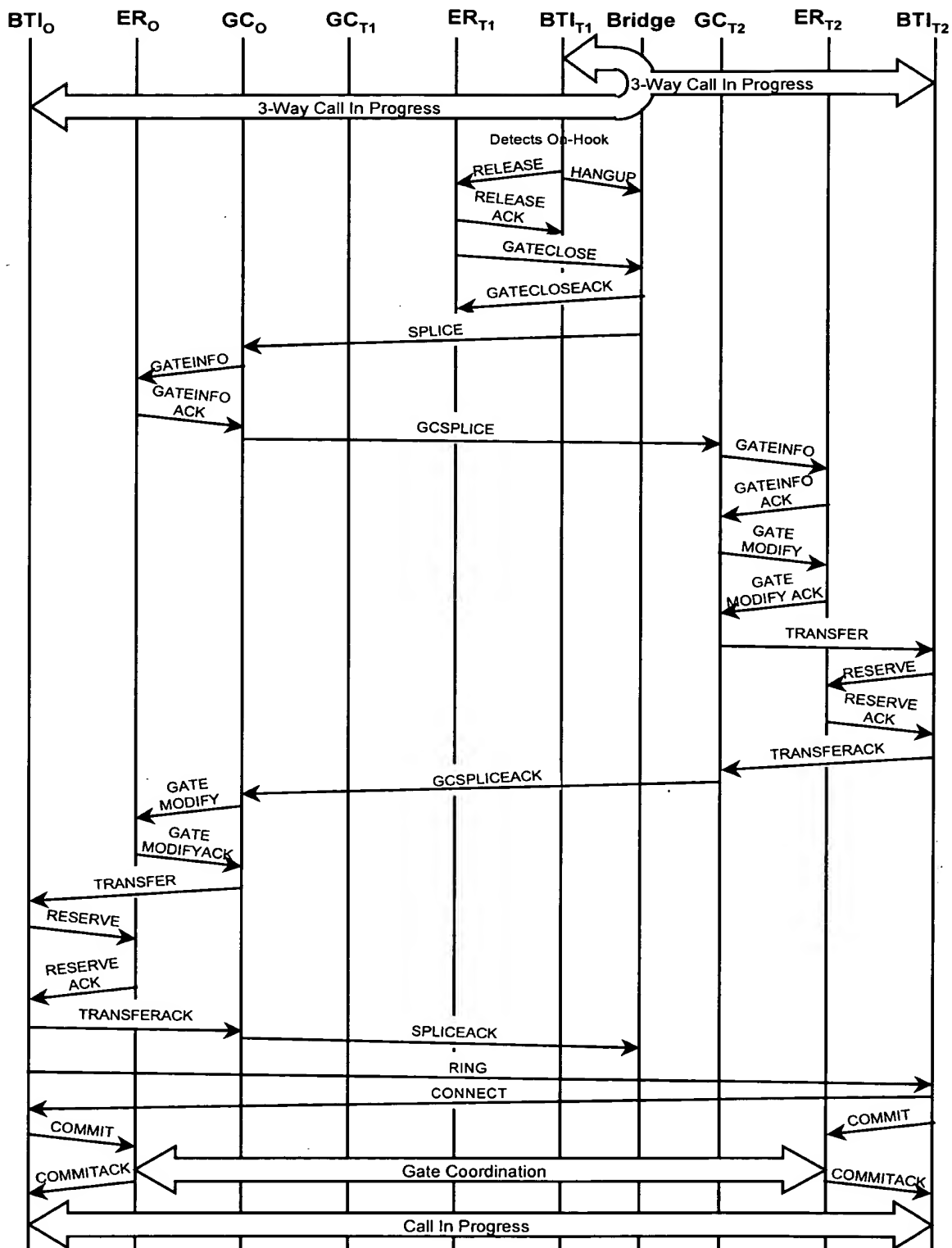


Figure 29

[illegible]

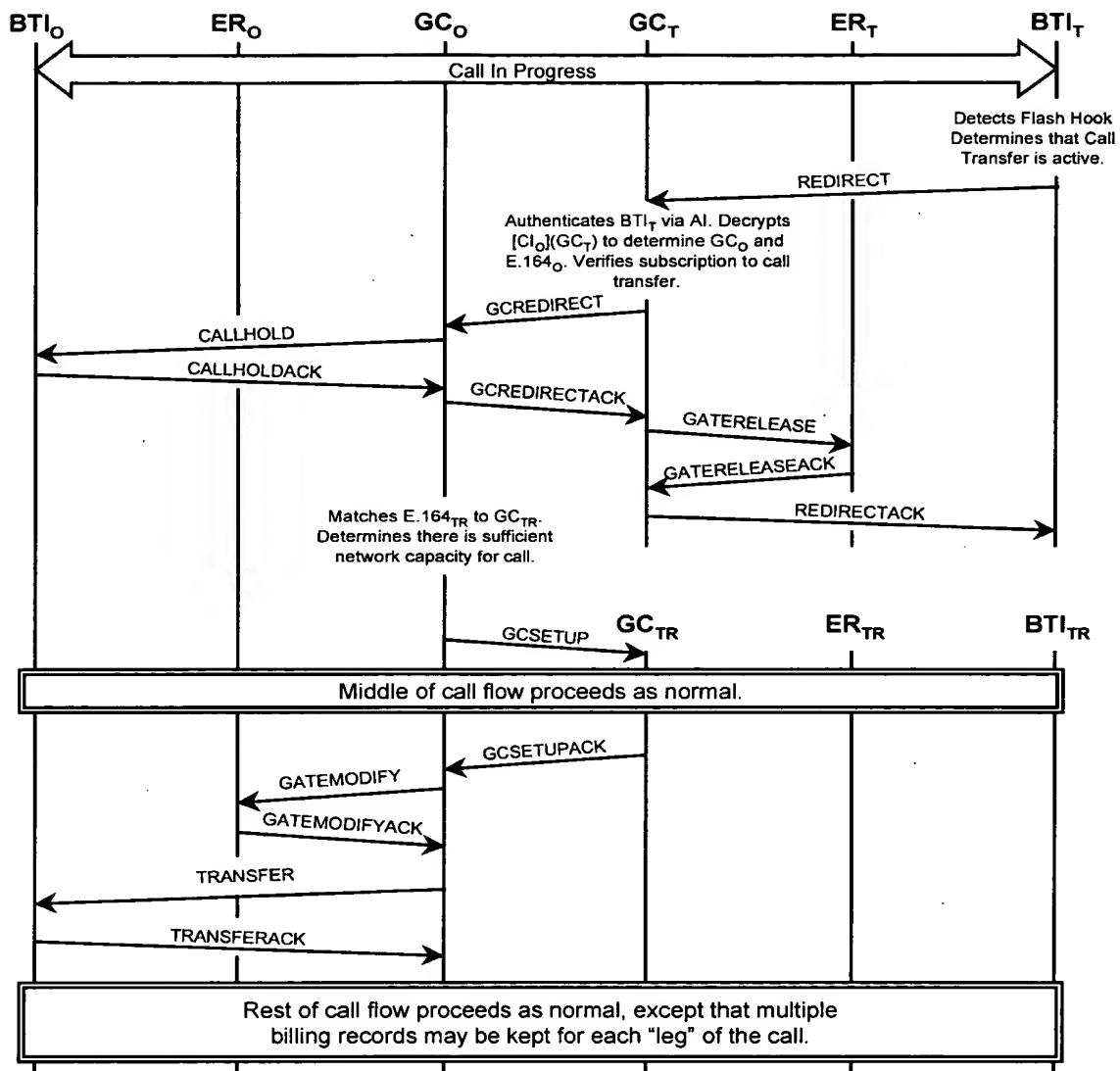


Figure 31

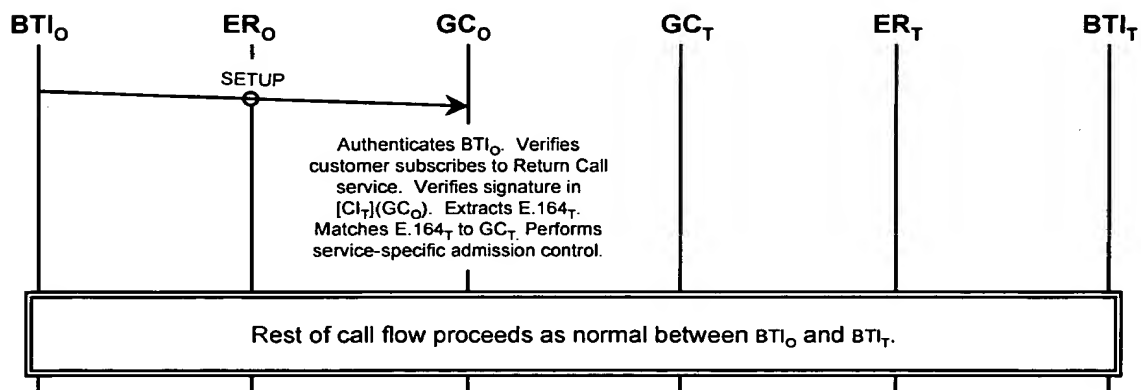


Figure 32